REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10/1/98	Final Technical Report 8/87-8/91

4. TITLE AND SUBTITLE
Microsupercomputers: Design & Implementation

5. FUNDING NUMBERS
N00014-87-K-0828

6. AUTHOR(S)
John L. Hennessy, Principal Investigator
Mark A. Horowitz, Co-Principal Investigator

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Stanford University
Terman Engineering, Room 214
Stanford, CA 94305

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 N. Fairfax
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This research project was to explore and develop the technologies necessary to build high performance computers from low-cost VLSI-based microprocessors. The project embraced a wide set of technologies for achieving these goals;
-Novel parallel computer architectures and associated parallel software
-High performance microprocessor design
-High speed CMOS and BiCMOS design
-Development of key supporting technologies such as testing and CAD technologies

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Microsupercomputers: Design and Implementation

Final Technical Report

Defense Advanced Research Projects Agency

Contract Number: N00014-87-K-0828

For the period of August 1987 - August 1991

Stanford University

Computer Systems Laboratory

Final Technical Report

Principal Investigator

Dr. John L. Hennessy

Associate Investigator

Dr. Mark A. Horowitz

August 1987 - August 1991

Contract No. N00014-87-K-0828

Order No. 1133, R & T Project Code: 4331685

This work is supported under the auspices of the Defense Advanced Research Projects Agency and the Office of Naval Research.

Overview

The goal of this research project was to explore and develop the technologies necessary to build high performance computers from low-cost VLSI-based microprocessors. The project embraced a wide set of technologies for achieving these goals, including:

- Novel parallel computer architectures and associated parallel software.
- High performance microprocessor design, including architecture, high speed component design (such as floating point units and caches), and novel compiler technologies.
- High speed CMOS and BiCMOS design.
- Development of key supporting technologies for designing high performance computers such as testing and CAD technologies.

Parallel Architecture

DASH-Directory Architecture for Shared Memory

The Stanford DASH multiprocessor advances the state of parallel computing by combining the programmability of shared-memory machines with the scalability of distributed-memory machines. The key idea on which DASH is built is that of distributed directory-based cache coherence; caches of the processors are kept coherent by sending point-to-point messages between processors on a scalable interconnection network

We designed and developed the DASH prototype and successfully demonstrated a 16 processor version using MIPS R3000/R3010 processors deliveringup to 400 MIPS of processing power. The interconnection network used consists of a pair of meshes, each with 16-bit wide channels using wormhole routing and derived from the Caltech routing chip.

DASH successfully demonstrates that scalable cache coherent shared memory architectures are possible. A large number of papers have been written about DASH and are cited at the end of this report. There is preliminary interest in industry to commercialize the DASH ideas.

Basic Parallel Architecture Studies

We completed several studies of invalidation patterns in multiprocessors. Our studies show that the best invalidation behavior is achieved when the cache line matches the size of the data objects being shared. Both line sizes that are too small and line sizes that are too large can drive up the average number of invalidations per shared write. We note, however, that this effect is not very strong. Consequently, it appears that high performance multiprocessors should use a large cache line size (32 - 64 bytes) to benefit from pre-fetching effects and to hide network latency encountered in scalable shared-memory multiprocessors.

Simulation Tools for Parallel Machines

We completed two novel systems for simulating multiprocessors. Tango is a software tracing and simulation system that provides data to aid in evaluating parallel programs and multiprocessor systems. The system provides a simulated multiprocessor environment by multiplexing application processes onto a single processor. Tango allows the user to trace the shared memory and synchronization behavior of parallel programs. The system is efficient, and can be used with a wide range of machine and programming models. The Tango system currently runs on the MIPS M120 and on the DECStation 3100. It is about 100-1000 times faster than older trace generators, and thus qualitatively changes the kinds of studies that we can do. In general, accurate tracing is difficult since parallel programs are typically non-deterministic; the execution path through the program depends on the real time behavior of the hardware system. Tango offers accurate tracing by allowing the user to optionally integrate his shared memory and synchronization timing simulations into the tracing environment. We currently have a detailed simulator of the DASH prototype integrated with Tango. This system is being extensively used for studying architectural tradeoffs and also for verifying our directory-based coherence protocol. A new system, called TangoLite, which is significantly faster was subsequently developed. Both Tango and TangoLite, as well as large amounts of program data were distributed to the parallel processing research community.

Parallel Software

The DASH Operating System

We developed an operating system for DASH and successfully booted it on the 16 processor DASH prototype. The OS provides users with the basic abstractions of UNIX System V with added functionality to make use of the new resources available in DASH. Our additions include the ability to attach processes to CPUs and clusters and the mechanisms to access the various novel locking and prefetching primitives provided by DASH. We have also extended the I/O subsystem to allow concurrent disk I/O.

Performance Debuggers for Parallel Programs

MTOOL is a tool for performance debugging of parallel programs. Case studies show that the information provided by MTOOL is extremely helpful: each of the scientific computing group's programs we have examined were sped up by 50-400% after one to two hours of work with MTOOL. A version of MTOOL was distributed for Silicon Graphics systems. A version of MTOOL was also used by Sun to develop a performance debugger for their machines.

Compiler Technology

We made major contributions to the development of compiler algorithm for loop-level optimizations to improve data locality in programs. These optimizations include loop interchange, reversal, skewing, and subblocking. Our technique is unique in that these optimizations are unified into a general transformation, thus we can find the best optimizations without trying every transformation sequence. This new approach to loop transformations can also be used in vectorizing and concurrentizing compilers, and industry has adopted our analysis technique

We also developed the first compiler system to integrate both high-level parallelizing transformations and scalar optimizations in a common framework. Existing parallelizing compilers use two separate programs: a source-to-source compiler performs high-level code restructuring, and a conventional compiler that translates source code into object code. The integration of parallelization and scalar optimizations is important because scalar information can be used for parallelization and vice versa. Many of the standard scalar optimizations such as constant propagation and forward substitution are useful for parallelism detection. Conversely, if we want to exploit instruction level parallelism, this high-level data dependence information must be made available to the machine code scheduler. Moreover, a clean, simple internal representation is more conducive to high level code transformations than the source level representation.

One of the key issues in integrating the parallelizing and scalar optimizations is the intermediate code representation. The former set of optimizations needs higher level information and the latter needs lower level information. The new intermediate format, called SUIF, captures this ability. We have gained some experience in using the intermediate format by building several rapid prototypes of code transformations.

Parallel Programming Languages

To compensate for the weakness of parallelizing compilers in extracting coarse-grain parallelism, we developed a parallel programming language called Jade. Starting with a sequential program, a programmer simply augments those sections of code to be parallelized with side effect information. The Jade system finds not just static parallelism but also parallelism that can only be derived at run time. Using Jade can significantly reduce the time and effort required to develop a parallel version of an imperative application with serial semantics. We have implemented a prototype Jade system, ported it to DASH, and programmed several applications using this language.

Uniprocessor Architecture

MIPS-X: Second Generation RISC

A major accomplishment during this contract was the completion and evaluation of MIPS-X, our second generation RISC architecture. In addition to reaching new performance levels, MIPS-X incorporates several novel architectural ideas:

- An architectural mechanism that allows integration of high performance coprocessors. Several commercial RISC designs have recently incorporated this approach.
- A novel mechanism for fast branching in pipelined machines, called squashing or canceling branches. Several companies have adopted this idea as well.
- A new technique for combining caches and TLBs to reduce cache access time.

The MIPS-X design was fully documented in a book; this book served as a "design manual" for a number of companies building RISC processors. In addition, the database was licensed to a number of companies that have used the processor core for application-specific processors.

Super-Scalar Design

We did extensive work on techniques for extracting instruction level parallelism in hardware, as well as in integrated hardware/software approaches. Our initial work on the non-scientific applications we are interested in, showed that it is possible to execute a program in about one-half the number of cycles needed by a conventional RISC machine. But, to achieve this type of speedup in hardware required branch prediction, a four wide instruction decoder, out-of-order execution, and register renaming; a non-trivial amount of hardware.

Our integrated approach used some new hardware structures together with sophisticated compiler scheduling technology. Our approach is to change the hardware slightly to allow the software to move code through branches, giving the compiler more opportunities to optimize the code. At present, we have the basic compiler up and running and have a system to estimate the performance advantage of moving code through branches. We also have a proposal for the basic machine organization.

Multi-level Caches

The presence of a second-level cache can decrease the optimum size and cycle time of the first-level cache, and significantly improve performance beyond the best attainable with a single level of caching. The optimal characteristics of the second level-cache depend on the miss ratio of the first-level cache, but in general, an optimal second level cache will be significantly larger and more likely to be associative than if it were alone in the system. This is because the presence of the first-level cache reduces the number of accesses to the second level without significantly reducing the number of misses in that cache. This shifts the tradeoff away from short cycle times and towards low miss ratios.

We performed novel new studies for the design of second-level caches which have been used by industry as they begin to incorporate second-level caches.

High-Speed Arithmetic

We have shown how to build dense, fast multipliers by building a partial multiplier tree, and pipelining the tree after every two carry-save adders. By clocking the structure quickly (100s of MHz) one can complete a multiply only slightly slower than a full tree. We also showed how to perform IEEE rounding in this type of structure. It turns out there is a way of starting the final carry propagate add before the carry-in is known. This result is very important since in iterative multipliers, the carry-in is only known 1 to 2 cycles after the rest of the result has been generated. Using this rounding method allows one to perform IEEE compatible multiplication nearly as fast as producing a truncated result. The overhead in hardware is also not too large, less than 25% added hardware.

High-speed VLSI Design, CAD, and Technology

BiCMOS Memory

During this period, we designed a 64K sRAM in TI's 0.8u BiCMOS technology. The sRAM was designed to demonstrate it is possible to build a large, high-speed (under 4ns) BiCMOS sRAM, while maintaining a reasonable power dissipation (1.5W). It uses the CSEA cell, with a bipolar transistor in each memory cell that we reported earlier. The design uses an innovative address decoder that combines the high-speed of a standard diode decoder, while greatly reducing the power that the decoder requires. The power reduction is accomplished by replacing the resistor load in a diode decoder with a pMOS device, and then using the gate of the pMOS to control its resistance. This technique allowed us to reduce the decoder power by a factor of 4, which reduced the power of the part by about a factor of 3. The sRAM also used a novel write-path. Instead of having a set of CMOS decoders for writes, the RAM uses the read decoders and provides a ECL-CMOS converter per wordline. Sharing the decoder increases the write speed, while reducing the complexity of the part.

The RAM is functional with an access time of 4ns. This BiCMOS memory cell design was used by a commercial company, Aspen, in their 3ns 4K BiCMOS RAM

We also designed a new BiCMOS CAM cell and built a TLB with a 4ns (pad to pad) translation delay. The 64-entry, fully-associative TLB was simply a demonstration vehicle for the CAM cell which uses small ECL-like swings to achieve new performance levels.

BiCMOS CAD

The last major part of our BiCMOS effort is in creating CAD tools to support BiCMOS designs. We are using Magic for layout, but simulation is a much more difficult problem. Trying to fill the current gap in simulation tools for BiCMOS circuits, we are working on Bisim; a Rsim like simulator for Bipolar and BiCMOS circuits. Like Rsim, Bisim is a switch level simulator. But unlike Rsim, Bisim understands that signals are voltages rather than simple Boolean values. This extra flexibility allows Bisim to handle a wider class of circuits than Rsim can handle. In particular it should be able to correctly simulate a wide class of sense circuits -- circuits that often occur in BiCMOS. At this point, we model both MOS and bipolar devices by piecewise linear devices and have written code that will find the final values for networks of these devices. These algorithms have been incorporated in Bisim, and we are now evaluating their performance by trying to simulate the BiCMOS sRAM we designed. The initial results look promising, but we have a significant amount of work still to do. Although we have simple timing models in this version of the simulator, it seems that these models might need some improving.

Low-Cost Testers

We developed a novel single chip tester. The chip, called Testarossa, contains a dRAM for the test vector storage, a decompressor to increase the effective vector size and the pin electronics for 16 DUT pins. The chip was fabricated in a 1.6m CMOS technology and used to construct a 256 pin, 25 MHz tester.

High Level Synthesis

We had a major breakthrough in optimal logic synthesis of digital synchronous sequential circuits. We have developed algorithms for minimizing the area of synchronous combinational and/or sequential circuits under cycle time constraints and the cycle time under area constraints. Previous approaches attacked this problem by separating the combinational logic from the registers and by applying circuit transformations to the combinational component only. We have shown instead how to optimize concurrently the circuit equations and the register position. This method is novel and achieves results that are at least as good as those obtained by previous methods. A computer implementation of the algorithms in program MINERVA has been accomplished and experimental results have supported the theory. This tool has been widely distributed to industry.

We also developed two tools for high level synthesis: Hercules and Hebe. The former performs behavioral level synthesis into an intermediate form that can be easily simulated. The latter performs user-directed structural synthesis and it embeds the implementation of the algorithms described before. The Hercules/Hebe tools have been used for two chip designs: a digital audio interface chip (CD or DAT to PC) and a discriminator of a multi-anode photodetector for the space telescope.

Protocol Verification

Formal verification of hardware is increasing in importance. We developed and applied new protocol verification techniques to the cache coherence protocol for DASH. We used a reduced description of the protocol which has been sufficient to find some cases that were not covered in the documentation of the protocol. We are continuing to enhance the DASH description and specification, with the goal of either proving it correct or accelerating the debuggingof the DASH prototype.

Parallel CAD Tools

Parallel simulation has been proposed to explore the potentials of the DASH multiprocessor machine for CAD applications. Integration of existing simulators (THOR, IRSIM, and SPICE) for mixed-mode circuit and system simulation has been developed to provide a parallel multi-level simulation environment. Parallelism is obtained within each simulator by decomposing its simulation into smaller blocks and among the simulators by providing a communication and synchronization interface between them.

We have also implemented the prototype on a network of DEC workstations and on an Intel iPSC/860 message-passing machine. Initial results show reasonable speed-ups for up to 16 processors.

We also developed a novel parallel place-and-route system called LocusRoute and successfully achieved speed-ups of up to 8.

Power and Noise Analysis

We completed and distributed Ariel, an analysis tool that allows the designer to calculate the noise on the power supply lines in the integrated circuit. This program first extracts the resistance of the supply lines, then estimates the supply current and finally uses this information to find the voltage drops.

3. Publications and Reports

1. Acken, J., Agarwal, A., Gulak, G., Horowitz, M., *et al.*, The MIPS-X RISC Microprocessor. 1 ed. P. Chow. 1989, Boston, MA: Kluwer Academic Publishers. 1-231.

2. Agarwal, A. and Gupta, A. "Memory Reference Characteristics of Multiprocessor Applications Under MACH" in *Sigmetrics*. ACM. May 1988.

3. Agarwal, A., Hennessy, J., and Horowitz, M., "Cache Performance for Operating Systems and Multiprogramming Workloads." ACM Transactions on Computers. Vol. **6**(4): pgs. 393-431. November 1988.

4. Agarwal, A., Simoni, R., Hennessy, J., and Horowitz, M. "Scalable Directory Schemes for Cache Consistency" in *15th International Symposium on Computer Architecture*. IEEE. Honolulu, HI. June 1988.

5. Agarwal, A. and Sites, R. "Multiprocessor Address Tracing and Characterization Using ATUM" in *15th International Symposium on Computer Architecture*. IEEE/ACM. Honolulu, HI. June 1988.

6. Agarwal, A., Horowitz, M., and Hennessy, J., "An Analytic Cache Model." Vol. 7(2): pgs. 184-215. May 1989.

7. Agarwal, A., Analysis of Cache Performance for Operating Systems and Multiprogramming. 1989, Kluwer Academic Publisher. 224.

8. Aho, A., Ganapathi, M., and Tjiang, S., "Code Generation Using Tree Matching and Dynamic Programming." October 1989.

9. Alur, R. and Dill, D.L. "Timed Finite Automata" in *Symposium on Foundations of Computer Science*. IEEE. 1989.

10. Alverson, R., Blank, T., and et. al. "THOR User's Manual". Stanford University. Technical Report, CSL-TR-88-349. January 1988.

11. Au, W.-Y. and Weise, D. "Automatic Generation of Compiled Simulations Through Program Specialization". June 1991.

12. Berlin, A. and Weise, D., "Compiling Scientific Code using Partial Evaluation." Vol. **23**(9): . December 1990.

13. Black, D., Gupta, A., and Weber, W.-D. "Competitive Management of Distributed Shared Memory" in *Compcon*. ACM. March 1989.

14. Blatt, M. "Yield Evaluation of a Soft-configurable WSI Switch Network" in *1989 IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*. Plenum Press. Tampa, Florida. October 1989.

15. Chandra, R., Gupta, A., and Hennessy, J.L. "COOL: A Language for Parallel Programming" in *Proceedings of the Second Workshop on Languages and Compilers for Parallel Computing*. Pitman. University of Illinois at Urbana-Champaign. August 1989.

16. Choi, K., Hwang, S.Y., and Blank, T. "Incremental-in-Time Algorithm for Digital Simulation" in *25th Design Automation Conference*. IEEE/ACM. Anaheim, CA. June 1988.

17. Chow, P. and Horowitz, M. "The Design and Testing of MIPS-X" in *Advanced Research in VLSI*. MIT Press. Cambridge, MA. March 1988.

18. Chow, F. and Hennessy, J., "The Priority-based Coloring Approach to Register Allocation." Transactions on Parallel Languagesand Systems. October 1990.

19. Davis, H. and Hennessy, J. "Characterizing the Synchronization Behavior of Parallel Programs" in *Symposium on Parallel Programming: Experience with Applications, Languages and Systems*. ACM. New Haven, CT. July 1988.

20. Davis, H. and Goldschmidt, S. "Tango: A Multiprocessor Simulation and Tracing System". Stanford University. Technical Report, CSL-90-436. July 1990.

21. Davis, H., Goldschmidt, S.R., and Hennessy, J. "Multiprocessor Simulation and Tracing Using Tango" in *International Conference on Parallel Processing (ICPP)*. St. Charles, IL. August 1991.

22. De Micheli, G. and Ku, D. "HERCULES- A System for High-Level Synthesis" in *Design Automation Conference*. IEEE/ACM. Anaheim, CA. June 1988.

23. De Micheli, G. and Klein, T. "Algorithms for Synchronous Logic Synthesis" in *International Symposium for Circuits and Systems*. IEEE. Portland, OR. May 1989.

24. Dill, D.L. "Trace Theory for Automatic Hierarchical Verification of Speed-independent Circuits" in *Advanced Research in VLSI: Proceedings of the Fifth MIT Conference*. MIT Press. 1988.

25. Dill, D.L. "Timing Assumptions and Verification of Finite-State Concurrent Systems" in *Workshop on Finite-State Concurrent Systems*. Grenoble, France. June 1989.

26. Dill, D.L., Nowick, S.M., and Sproull, R.F. "Automatic Verification of Speed-Independent Circuits With Petri Net Specifications" in *IEEE International Conference on Computer Design*. IEEE Computer Society. 1989.

27. Dill, D. and Loewenstein, P. "Verification of Multiprocessor Cache Protocol using Refinement Relations and Higher-Order Logic" in *Workshop on Computer-Aided Verification*. Rutgers University. 1990.

28. Dill, D. and Loewenstein, P. "Formal Verification of Cache Systems using Refinement Relations" in *International Conference on Computer Design*. IEEE. pgs. 228-233. 1990.

29. Ganapathi, M., "Prolog Based Retargetable Code Generation." Computing Languages Vol. **14**(3): pgs. 193-204. February 1989.

30. Ganapathi, M. and Kennedy, K. "Interprocedural Analysis and Optimization". Rice University. Technical Report, COMP TR 89-96. July 1989.

31. Ganapathi, M., "Semantic Predicates in Parser Generators." Computing Languages Vol. **14**(1): pgs. 25-33. October 1989.

32. Ganapathi, M. and Mendal, G., "Issues in Ada Compiler Technology." Vol. **22**(2): pgs. 52-60. February 1989.

33. Gasbarro, J. and Horowitz, M. "Integrated Pin Electronics for VLSI Functional Testers" in *Custom Integrated Circuits Conference*. IEEE. Rochester, NY. pgs. 16.2.1-16.2.4. May 1988.

34. Gasbarro, J. and Horowitz, M., "Integrated Pin Electronics for VLSI Functional Testers." Journal of Solid-State Circuits. Vol. **SC-24**(2): pgs. 331-337. April 1989.

35. Gasbarro, J. and Horowitz, M. "Testarossa: A Single-Chip, Functional Tester for VLSI Circuits" in *International Solid-State Circuits Conference*. IEEE. San Francisco, CA. February 1990.

36. Gharachorloo, K., Sarkar, V., and Hennessy, J. "A Simple and Efficient Implementation Approach for Single Assignment Languages" in *Lisp and Functional Programming Conference*. ACM. Salt Lake City, UT. July 1988.

37. Gharachorloo, K., Lenoski, D., Laudon, J., Gupta, A., *et al.* "Memory Consistency and Event Ordering in Scalable Shared-Memory Multiprocessors" in *17th Annual International Symposium on Computer Architecture*. IEEE/ACM. Seattle, WA. May 1990. Also appears as Stanford University technical report number CSL-89-405, published December 1989.

38. Gharachorloo, K., Gupta, A., and Hennessy, J.L. "Performance Evaluation of Memory Consistency Models for Shared-Memory Multiprocessors" in *Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IV)*. ACM/IEEE. Santa Clara, CA. April 1991.

39. Gharachorloo, K., Gupta, A., and Hennessy, J. "Two Techniques to Enhance the Performance of Memory Consistency Models" in *International Conference on Parallel Processing (ICPP)*. St. Charles, IL. August 1991. Also published as Technical Report No. CSL-TR-91-473.

40. Gharachorloo, K. and Gibbons, P. "Detecting Violations of Sequential Consistency" in *3rd Symposium on Parallel Algorithms and Architectures (SPAA)*. ACM. Hilton Head, SC. pgs. July 1991. Also published as Technical Report No. CSL-TR-91-474.

41. Gharachorloo, K., Adve, S., Gupta, A., Hennessy, J., *et al.*, "Programming for Different Memory Consistency Models." 1991.

42. Gibbons, P.B., *A Synthesis of Parallel Algorithms*, in *Asynchronous PRAM Algorithms* J.H. Reif. 1990, Morgan-Kaufmann: San Mateo.

43. Gibbons, P.B. "Cache Support for the Asynchronous PRAM" in *19th International Conference on Parallel Processing*. ACM. St. Charles, IL. August 1990.

44. Goldberg, A. and Hennessy, J. "MTOOL: A Method for Isolating Memory Bottlenecks in Shared Memory Multiprocessor Programs" in *International Conference on Parallel Processing (ICPP)*. St. Charles, IL. August 1991.

45. Goldberg, A. and Hennessy, J. "Performance Debugging Shared Memory Multiprocessor Programs with MTOOL" in *Supercomputing 91*. IEEE/ACM. Albuquerque, NM. November 1991.

46. Goldschmidt, S. and Davis, H. "Tango Introduction and Tutorial". Stanford University, Computer Systems Laboratory. Technical Report, CSL-90-410. January 1990.

47. Gopinath, K. "Copy Elimination with Abstract Interpretation". Stanford University. Computer Science Department Report, Classic 87-17. February 1988.

48. Gupta, A. and Forgy, C., "Static and Run-Time Characteristics of Production Systems." June 1987.

49. Gupta, A., Forgy, C., Kalp, D., Newell, A., *et al.*, "Parallel Implementation of OPS5 on the Encore Multiprocessor: Results and Analysis." Vol. **17**(2): 1988.

50. Gupta, A. and Tucker, A. "Exploiting Variable Grain Parallelism at Runtime" in *Symposium on Parallel Programming: Experience with Applications, Languages, and Systems*. ACM. New Haven, CT. July 1988.

51. Gupta, A. and Tambe, M. "Suitability of Message Passing Computers for Implementing Production Systems" in *AAAI-88*. St. Paul, MN. August 1988.

52. Gupta, A., Forgy, C.L., Kalp, D., Newell, A., *et al.* "Parallel OPS5 on the Encore Multimax" in *Proceedings of International Conference on Parallel Processing*. 1988.

53. Gupta, A., Forgy, C., and Newell, A., "High-Speed Implementations of Rule-Based Systems." Vol. 7(2): pgs. 119-146. May 1989.

54. Gupta, A., Weber, W.-D., and Mowry, T. "Reducing Memory and Traffic Requirements for Scalable Director-Based Cache Coherence Schemes" in *International Conference on Parallel Processing*. Pennsylvania State University. pgs. 312-321. August 1990. Also appears as Stanford University Technical Report CSL-90-417, March 1990.

55. Gupta, A. and Weber, W.-D., "Cache Invalidation Patterns in Shared-Memory Multiprocessors." 1991.

56. Gupta, A., Hennessy, J., Gharachorloo, K., Mowry, T., *et al.* "Comparative Evaluation of Latency Reducing and Tolerating Techniques" in *18th International Symposium on Computer Architecture (ISCA)*. IEEE/ACM. Toronto, Canada. pgs. 254-263. May 1991. Also appears as Stanford Univ. Technical Report No. CSL-TR-91-467, March 1991.

57 Gupta, A., Tucker, A., and Stevens, L. "Making Effective Use of Shared-Memory Multiprocessors: The Process Control Approach". Stanford University, Computer Systems Laboratory. Technical Report, CSL-TR-91-475. May 1991.

58. Gupta, A., Tucker, A., and Urushibara, S. "Impact of Operating System Scheduling Policies and Synchronization Methods on the Performance of Parallel Applications" in *Proceedings of SIGMETRICS*. ACM. pgs. 120-132. May 1991.

59. Hayes, B. "Anonymous One-Time Signatures and Flexible Untracable Electronic Cash" in *Proceedings of AUSCRYPT*. Sydney, Australia. pgs. 228-238. January 1990. To appear.

60. Hennessy, J.L., "Beyond RISC." Unix Review. Vol. 7(9): pgs. 48-57. September 1989.

61. Hennessy, J.L. "RISC Architecture: A Perspective on the Past and Future" in *Decennial Caltech VLSI Conference*. The MIT Press. Pasadena, CA. pgs. 37-42. March 1989. Invited Presentation.

62. Hennessy, J.L. and Patterson, D.A., Computer Architecture: A Quantitative Approach. 1st ed. 1990, San Mateo, CA: Morgan Kaufmann Publishers, Inc. 1-659.

63. Hennessy, J.L. and Jouppi, N.P., "Computer Technology and Architecture: An Evolving Interaction." IEEE Computer. Vol. **24**(9): pgs. 18-29. September 1991. Special 40th Anniversary Issue.

64. Horowitz, M. and al, e., "MIPS-X: A High-Speed 32 Bit Processor." Journal of Solid-State Circuits. Vol. **SC-22**(5): pgs. 790-799. October 1987.

65. Horowitz, M., Slamowitz, M., Rose, B., and Johnson, M. "A 3.5ns, 1 Watt, ECL Register File" in *International Solid-State Circuits Conference*. IEEE. San Francisco, CA. February 1990.

66. Hwang, S.Y., Blank, T., and Choi, K., "Fast Functional Simulation: An Incremental Approach." Computer Aided Design Vol. **CAD-7**(7): pgs. 765-774. July 1988.

67. Johnson, W. "Super-Scalar Processor Design." Ph.D. Thesis from Stanford University, 1989.

68. Kao, R., Alverson, R., Horowitz, M., and Stark, D. "Bisim: A Simulator for Custom ECL Circuits" in *International Conference on Computer-Aided Design*. IEEE. Santa Clara, CA. pgs. 62-65. November 1988.

69. Kao, R. and Horowitz, M. "Efficient Moment-Based Timing Analysis for Variable Accuracy Switch Level Simulation". Stanford University. Technical Report, CSL-54-91-468. April 1991.

70. Kao, R. and Horowitz, M. "Asymptotic Waveform Evaluation for Circuits With Redundant DC Equations". Stanford University, Computer Systems Laboratory. Technical Report, CSL-TR-91-478. May 1991.

71. Katz, M. and Weise, D. "Continuing into the Future: On the Interaction of Future and First-Class Continuations" in *Joint US/Japan Workshop on Parallel Lisp*. Sendai, Japan. June 1989.

72. Katz, R.H. and Hennessy, J.L., "High Performance Microprocessor Architectures." International Journal of High Speed Electronics. Vol. **1**(1): pgs. 1-18. March 1990.

73. Katz, M. and Weise, D. "Continuing into the Future: On the Interaction of Futures and First-Class Continuations" in *Conference on Lisp and Functional Programming*. ACM. Nice, France. pgs. 176-184. June 1990.

74. Ku, D. and De Micheli, G. "Hardware C - A Languagefor Hardware Design". Stanford University. Technical Report, CSL 88-362. August 1988.

75. Kung, H.T. and Lam, M. "An Approach to Automatic Generation of Linear Systolic Array Programs" in *International Conference and Exhibition on Parallel Processing for Computer Vision and Display*. London, England. January 1988.

77. Lam, M., A Systolic Array Optimizing Compiler. 1988, Kluwer Academic Publishers.

78. Lam, M. "Software Pipelining: An Effective Scheduling Technique for VLIW Machines" in *SIGPLAN Conference on Programming Language Design and Implementation*. ACM. Atlanta, GA. pgs. 318-328. June 1988.

79. Lam, M. "Compiler Optimizations for Asynchronous Systolic Array Programs" in *15th Annual Symposium on Principles of Programming Languages*. ACM. San Diego, CA. pgs. 309-318. January 1988.

80. Lam, M. "Compiler Optimizations for Superscalar Computers" in *9th International Conference on Computing Methods in Applied Sciences and Engineering*. Paris, France. January 1990.

81. Lam, M., "Instruction Schedulingfor Superscalar Architectures." Annual Review. Vol. 4: pgs. 173-201. 1990.

82. Lam, M., "The Software Pipelining Algorithm and Experimental Results." Transactions on Parallel Languagesand Systems. 1990.

83. Lam, M.S. and Rinard, M.S. "Coarse-Grain Parallel Programming in Jade" in *Third Symposium on Principles and Practice of Parallel Programming (PPoPP '91).* ACM Sigplan. Williamsburg, VA. pgs. 94-105. April 1991.

84. Lam, M.S., Rothberg, E.E., and Wolf, M.E. "The Cache Performance and Optimizations of Blocked Algorithms" in *Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IV).* Santa Clara, CA. April 1991.

85. Lam, M.S. and Wolf, M.E. "Automatic Blocking by a Compiler" in *Fifth SIAM Conference on Parallel Processing for Scientific Computing.* March 1991.

86. Lenoski, D., Gharachorloo, K., Laudon, J., Gupta, A., *et al.* "Design of the Stanford DASH Multiprocessor". Stanford University, Computer Systems Laboratory. Technical Report, CSL-89-403. December 1989.

87. Lenoski, D., Laudon, J., Gharachorloo, K., Gupta, A., *et al.* "The Directory-Based Cache Coherence Protocol for the DASH Multiprocessor" in *17th Annual International Symposium on Computer Architecture.* IEEE. Seattle, WA. pgs. 148-159. May 1990. Also appears as Stanford University technical report number CSL-89-404 published December 1989.

88. Lenoski, D., Gharachorloo, K., Laudon, J., Gupta, A., *et al.* "Design of Scalable Shared-Memory Multiprocessors: The DASH Approach" in *Compcon.* ACM. February 1990.

89. Lenoski, D., Laudon, J., Gharachorloo, K., Weber, W.-D., *et al.* "Overview and Status of the Stanford DASH Multiprocessor" in *ISSMM Conference.* Tokyo, Japan. April 1991.

90. Lenoski, D., Laudon, J., Gharachorloo, K., Weber, W.-D., *et al.*, "The Stanford DASH Multiprocessor." IEEE Computer 1991.

91. Linton, M.A., Vlissides, J.M., and Calder, P.R., "Composing User Interface with InterViews." IEEE Computer. February 1989.

92. Martonosi, M. and Gupta, A. "Tradeoffs in Message Passing and Shared Memory Implementations of a Standard Cell Router" in *International Conference on Parallel Processing.* pgs. 88-96. August 1989.

93. Martonosi, M. and Gupta, A., "MemSpy: Analyzing Memory System Bottlenecks in Programs." 1991.

94. Maydan, D.E., Hennessy, J.L., and Lam, M.S. "Efficient and Exact Data Dependence Analysis" in *ACM SIGPLAN 91 Conference on Programming Language Design and Implementation*. Toronto, Ontario, Canada. pgs. 1-14. June 1991.

95 McFarling, S. "Program Optimization for Instruction Caches" in *ASPLOS-III*. ACM/IEEE. Boston, MA. April 1989.

96. McFarling, S. "Procedure Merging with Instruction Caches" in *Conference on Programming Language Design and Implementation (PLDI)*. ACM Sigplan. Toronto, Canada. June 1991.

97. Mowry, T. and Gupta, A., "Tolerating Latency Through Software-Controlled Prefetching in Scalable Shared-Memory Multiprocessors." Journal of Parallel and Distributed Computing. Vol. **12**(2): pgs. 87-106. June 1991.

98. Nayak, P., Gupta, A., and Rosenbloom, P. "Comparison of Rete and Treat Production Matchers for Soar" in *AAAI-88*. St. Paul, MN. August 1988.

99. Nguyen, T.A. and Weise, D. "Diagnosing Multiple Faults in Digital Systems." in *Artificial Intelligence Applications Conference*. IEEE. Miami Beach, FL. pgs. 151-158. March 1989.

100. Nowick, S.M. and Dill, D.L. "Practicality of State-Machine Verification of Speed-independent Circuits" in *International Conference on Computer-Aided Design*. 1989..

101. Odani, M., Hwang, S.Y., Blank, T., and Rokicki, T., "The Hermod Behavioral Synthesis System." Journal of Systems and Software 1989.

102. Okuno, H. and Gupta, A. "Parallel Execution of OPS5 in QLISP" in *4th IEEE Conference on Artificial Intelligence Applications*. IEEE. March 1988.

103. Przybylski, S., Horowitz, M., and Hennessy, J. "Performance Tradeoffs in Cache Design" in *15th International Symposium on Computer Architecture*. IEEE. Honolulu, HI. pgs. 290-298. May 1988.

104. Przybylski, S. "Performance-Directed Memory Hierarchy Design." Ph.D. Thesis from Stanford University, 1988.

105. Przybylski, S., Horowitz, M., and Hennessy, J. "Characteristics of Performance-Optimal Multi-Level Cache Hierarchies" in *16th International Symposium on Computer Architecture*. IEEE/ACM. Jerusalem, Israel. June 1989.

106. Richardson, S. and Ganapathi, M. "Interprocedural Analysis Vs. Procedure Integration" in Information Processing Letters. April 1989.

107. Richardson, S. and Ganapathi, M., "Code Optimization Across Procedures." IEEE Computer. Vol. **22**(2): pgs. 42-50. February 1989.

108. Richardson, S. and Ganapathi, M., "Interprocedural Optimization: Experimental Results." Software -- Practice and Experience. Vol. 19(2): pgs. 149-169. February 1989.

109. Richardson, S.E. "Evaluating Interprocedural Code Optimization Techniques." Ph.D. Thesis from Stanford University, February 1991. Also published as Technical Report No. CSL-91-460.

110. Rinard, M.C. and Lam, M.S. "Semantic Foundations of Jade" in *19th Annual Symposium on Principles of Programming Languages*. ACM. January 1992.

111. Rose, J.S. "The Parallel Decomposition and Implementation of an Integrated Circuit Global R" in *Sigplan Symposium on Parallel Programming*. ACM. New Haven, CT. pgs. 138-145. July 1988.

112. Rose, J.S. "LocusRoute: A Parallel Global Router for Standard Cells" in *25th Design Automation Conference*. IEEE/ACM. Anaheim, CA. pgs. 189-195. June 1988. An abstract version also appeared in the MCNC, Workshop on Placement and Routing, Atlanta, GA, May 1988.

113. Rose, J.S., Klebsch, W., and Wolf, J. "Equilibrium Detection and Temperature Measurement of Simulated Annealing Placements" in *Workshop on Placement and Routing*. MCNC. Atlanta, GA. May 1988.

114. Rose, J., Klebsch, W., and Wolf, J. "Temperature Measurement of Simulated Annealing Placements" in *International Conference on Computer-Aided Design*. IEEE. pgs. 514-517. 1988.

115. Rose, J., "Parallel Global Routing for Standard Cells." IEEE Transactions on Computer-Aided Design of Circuits and Systems. 1989.

116. Rose, J., Klebsch, W., and Wolf, J., "Temperature Measurement and Equilibrium Dynamics of Simulated Annealing Placements." IEEE Trans. on Computer-Aided Design of Circuits and Systems. 1989.

117. Rose, J., Francis, R., Chow, P., and Lewis, D. "The Effect of Logic Block Complexity on Area of Programmable Gate Arrays" in *Custom Integrated Circuits Conference*. IEEE. 1989.

118. Roseel, G., Horowitz, M., Cline, R., and Dutton, R. "A Single-Ended BiCMOS Sense Circuit for Digital Circuits" in *International Solid State Circuits Conference*. IEEE. February 1989.

119. Rothberg, E. and Gupta, A. "Experiences Implementing a Parallel ATMS on a Shared-Memory Multiprocessor" in *International Joint Conference on Artificial Intelligence*. August 1989.

120. Rothberg, E. and Gupta, A. "Fast Sparse Matrix Factorization on Modern Workstations". Stanford University. Technical Report, October 1989.

121. Rothberg, E. and Gupta, A. "A Comparative Evaluation of Nodal and Supernodal Parallel Sparse Matrix Factorization: Detailed Simulation Results". Stanford University, Computer Systems Laboratory. Technical Report, CSL-90-416. February 1990. Also appears as STAN-CS-90-1305 published under the auspices of the Computer Science Department.

122. Rothberg, E. and Gupta, A., "Efficient Sparse Matrix Factorization on High-Performance Workstations--Exploiting the Memory Hierarchy." ACM Transactions on Mathematical Software. 1990.

123. Rothberg, E. and Gupta, A. "Parallel ICCCG on a Hierarchical Memory Multiprocessor--Addressing the Triangular Solve Bottleneck". Stanford University, Computer Systems Laboratory. Technical Report, CSL-90-449. September 1990.

124. Rothberg, E. and Gupta, A. "Techniques for Improving the Performance of Sparse Matrix Factorization on Multiprocessor Workstations" in *Supercomputing '90*. IEEE Computer Society. New York , NY. November 1990.

125. Rothberg, E. and Gupta, A. "An Evaluation of Left-Looking, Right-Looking, and Multifrontal Approaches to Sparse Cholesky Factorization on Hierarchical-Memory Machines". Stanford University. Technical Report, CSL-91-487/STAN-CS-91-1377. August 1991.

126. Ruf, E. and Weise, D. "Nondeterminism and Unification in LogScheme:Integrating Logic and Functional Programming" in *Fourth Conference on Functional Programming Languages and Computer Architecture*. ACM. London, England. pgs. 327-339. September 1989.

127. Ruf, E. and Weise, D. "LogScheme: Integrating Logic Programminginto Scheme" in *Lisp and Symbolic Computation*. 1990.

128. Ruf, E. and Weise, D. "Using Types to Avoid Redundant Specializations" in *ACM Sigplan Symposium on Partial Evaluation of Semantics Based Program Manipulation*. June 1991.

129. Salz, A. and Horowitz, M. "IRSIM: An Incremental MOS Switch-Level Simulator" in *26th Design Automation Conference*. IEEE/ACM. June 1989.

130. Santoro, M. and Horowitz, M., "SPIM: A Pipelined 64 X 64 Bit Interative Multiplier." Journal of Solid-State Circuits. Vol. **SC-24**(2): pgs. 487-493. April 1989. Also presented at the 1988 ISSCC in San Francisco, CA, February 1988.

131. Santoro, M. "Design and Clocking of VLSI Multipliers." Ph.D. Thesis from Stanford University, October 1989.

132. Santoro, M., Bewick, G., and Horowitz, M. "Rounding Algorithms for IEEE Multipliers" in *9th Symposium on Computer Arithmetic*. IEEE. Santa Monica, CA. pgs. 176-183. September 1989.

133. Saraswat, V., Rinard, M., and Panangaden. "Semantic Foundations of Concurrent Constraint Programming" in *Proceedings of the 18th Annual Symposium on Principles of Programming Languages*. ACM. January 1991.

134. Sarkar, V. "Partitioning and Scheduling Parallel Programs for Execution on Multiprocessors." Ph.D. Thesis from Stanford University, 1987. Also published under Technical Report CSL-TR-87-328

135. Scales, D., Rinard, M., Lam, M., and Anderson, J. "Hierarchical Concurrency in Jade" in *Fourth Workshop on Languages and Compilers for Parallel Computing*. August 1991.

136. Schnorf, P. and Ganapathi, M. "Compilation of Single Assignment Languages: Analysis and Propositions". Stanford University. Technical Report, CSL-89-399. November 1989.

137. Simoni, R. and Horowitz, M. "Modeling the Performance of Limited Pointers Directories for Cache Coherence" in *18th International Symposium on Computer Architecture*. IEEE/ACM. Toronto, Canada. pgs. 308-318. May 1991.

138. Simoni, R. and Horowitz, M. "Dynamic Pointer Allocation for Scalable Cache Coherence Directories" in *International Symposium on Shared Memory Multiprocessing*. Tokyo, Japan. pgs. 72-81. April 1991. Also appears as a Technical Report at Stanford University, Computer Systems Laboratory, CSL-TR-491.

139. Singh, J.P. and Hennessy, J.L., "Parallelizing the Simulation of Ocean Eddy Currents." Journal of Parallel and Distributed Computing. 1991. Also published as Stanford technical report, CSL-TR-89-388.

140. Singh, J.P. and Hennessy, J.L. "Automatic and Explicit Parallelization of an N-Body Simulation" in *TENCON '91*. IEEE. 1991. Also published as Stanford University technical report, number CSL-TR-91-462.

141. Singh, J.P. and Hennessy, J.L. "An Empirical Investigation of the Effectiveness and Limitations of Automatic Parallelization" in *ISSMM Conference*. Tokyo, Japan. April 1991.

142. Singh, J.P., Weber, W.-D., and Gupta, A. "SPLASH: Stanford Parallel Applications for Shared-Memory" in *Workshop of the International Symposium on Computer Architecture*. IEEE/ACM. Toronto, Canada. May 1991. Also published as Stanford Technical Report Number CSL-91-469.

143. Singh, J.P. and Hennessy, J.L. "Data Locality and Memory System Performance in the Parallel Simulation of Ocean Eddy Currents" in *Second Symposium on High Performance*

Computing. Cedex, France. October 1991. Also published as Technical Report CSL-TR-91-490.

144. Singh, J.P. and Hennessy, J.L., "Finding and Exploiting Parallelism in an Ocean Simulation Program: Experience, Results and Implications." Journal of Parallel and Distributed Computing 1991.

145. Singh, J. and Hennessy, J.L., "Parallelizing an Ocean Simulation Program: Experience, Results and Implications." Journal of Parallel and Distributed Computing 1991.

146. Smith, M., Johnson, M., and Horowitz, M. "Limits on Multiple Instruction Issue" in *ASPLOS-III*. ACM/IEEE. Boston, MA. pgs. 290-302. April 1989. Also appears as Stanford University Technical Report CSL-90-433 published July 1990.

147. Smith, M., Lam, M., and Horowitz, M. "Boosting Beyond Static Scheduling in a Superscalar Processor" in *17th Annual International Conference on Computer Architecture*. IEEE/ACM. Seattle, WA. pgs. 344-354. May 1990. Also appears as Stanford University Technical Report CSL-90-434 published July 1990.

148. Soule, L. and Blank, T. "Parallel Logic Simulation on General Purpose Machines" in *25th Design Automation Conference*. IEEE/ACM. Anaheim, CA. June 1988.

149. Soule, L. and Gupta, A. "Characterization of Parallelism and Deadlocks in Distributed Logic Simulation" in *26th Design Automation Conference*. IEEE/ACM. Las Vegas, NV. June 1989.

150. Soule, L. and Gupta, A. "Analysis of Parallelism and Deadlocks in Distributed-Time Logic Simulation". Stanford University, Computer Systems Laboratory. Technical Report, CSL-89-378. May 1989.

151. Soule, L. and Gupta, A., "Parallel Distributed-Time Logic Simulation." IDTOC. pgs. 32-48. December 1989.

152. Soule, L. "Parallel Logic Simulation: An Evaluation of Centralized-Time and Distributed-Time Algorithms." Ph.D. Thesis from Stanford University, 1991.

153. Soule, L. and Gupta, A., "An Evaluation of the Chandy-Misra-Bryant Algorithm for Digital Logic Simulation." ACM Transactions on Modeling and Computer Simulation October 1992.

154. Stark, D. and Horowitz, M. "Analyzing CMOS Power Supply Networks Using Ariel" in *25th Design Automation Conference*. IEEE/ACM. Anaheim, CA. June 1988.

155 Stark, D. and Horowitz, M., "Techniques for Calculating Currents and Voltages in VLSI Power Supply Networks." Transactions on Computer-Aided Design. Vol. 9(2): pgs. 126-132. February 1990.

156. Stark, D. "Analysis of Power Supply Networks in VLSI Circuits." Ph.D. Thesis from Stanford University, March 1991. Also published as Technical Report No. CSL-TR-91-465

157. Swami, A. and Gupta, A. "Optimization of Large Joint Queries" in *SIGMOD*. ACM. June 1988.

158. Tjiang, S.W.K., Wolf, M.E., Lam, M.S., Pieper, K., *et al.* "Integrating Scalar Optimizations and Parallelization" in *Fourth Workshop on Languages and Compilers for Parallel Computing*. August 1991.

159. Todesco, A. and Meng, T. "Interval Methods for Distributed Simulation systems" in *Proceedings on SCS Multiconference on Modeling and Simulation on Microcomputers*. San Diego, CA. January 1990.

160. Torrellas, J., Hennessy, J., and Weil, T. "Analysis of Critical Architectural and Program Parameters in a Hierarchical Shared-Memory Multiprocessor" in *Sigmetrics*. ACM. Denver, CO. May 1990.

161. Torrellas, J. and Hennessy, J. "Estimating the Performance Advantages of Relaxing Consistency in a Shared Memory Multiprocessor". Stanford University, Computer Systems Laboratory. Technical Report, CSL-90-365. February 1990.

162. Torrellas, J., Lam, M., and Hennessy, J.L. "Shared Data Placement Optimizations to Reduce Multiprocessor Cache Miss Rates" in *International Conference on Parallel Processing*. Pennsylvania State University. pgs. 266-270. August 1990.

163. Torrellas, J., Lam, M., and Hennessy, J.L., "Measurement, Analysis, and Improvement of the Cache Behavior of Shared Data in Cache Coherent Multiprocessors." ACM Transactions on Computer Systems. 1991..

164. Tucker, A. and Gupta, A. "Process Control and Scheduling Issues for Multiprogrammed Shared-Memory Multiprocessors" in *12th Symposium on Operating Systems Principles*. ACM. Lichtfield Park, AZ. December 1989.

165. Tucker, A., Stevens, L., and Gupta, A. "Making Effective Use of Shared-Memory Multiprocessors: The Process Control Approach" in *USENIX Winter Technical Conference*. 1991. Also published as Stanford Technical Report Number CSL-91-475.

166. Weber, W.D. and Gupta, A. "Exploring the Benefits of Multiple Hardware Contexts in a Multiprocessor Architecture: Preliminary Results" in *16th International Symposium on Computer Architecture*. IEEE. Jerusalem, Israel. June 1989.

167. Weber, W.D. and Gupta, A. "Analysis of Cache Invalidation Patterns in Multiprocessors" in *ASPLOS-III*. IEEE/ACM. Boston, MA. April 1989.

168. Weise, D. "Constraints and Multilevel Verification" in *Workshop on Hardware Specification, Verification and Synthesis: Mathematical Aspects*. Ithaca, NY. July 1989.

169. Weise, D. and Seligman, S., *General Compiled Electrical Simulation*. 1989,

170. Weise, D. "Constraint Posting for Verifying VLSI Circuits" in *Eleventh International Joint Conference on Artificial Intelligence*. ACM. Detroit, MI. pgs. 881-886. August 1989.

171. Weise, D., "Formal Verification of MOS Circuits." Transactions on Computer-Aided Design. Vol. **9**(4): pgs. 341-351. 1990.

1725. Weise, D. "Graphs as an Intermediate Representation for Partial Evaluation". Stanford University, Computer Systems Laboratory. Technical Report, CSL-90-421. March 1990.

173. Weise, D. and Ruf, E. "Computing Types During Program Specialization". Stanford University, Computer Systems Lab. Technical Report, CSL-TR-90-441. October 1990.

174. Williams, T.E., Horowitz, M., Alverson, R.L., and Yang, T.S. "A Self-Timed Chip for Division" in *Advanced Research in VLSI*. MIT Press. Stanford, CA. 1987.

175. Williams, T.E. and Horowitz, M.A., "Bipolar Circuit Elements Providing Self-Completion-Indication." IEEE Journal of Solid State Circuits Vol. **25**(1): pgs. 309-312. February 1990.

176. Williams, T.E. "Latency and Throughput Tradeoffs in Self-Timed Asynchronous Pipelines and Rings". Stanford University, Computer Systems Laboratory. Technical Report, CSL-90-431. May 1990.

177. Williams, T.E. and Horowitz, M.A. "A 160nS Division Implementation Using Self-Timing and Symmetric Overlapped Execution" in *IEEE Conference on Computer Arithmetic (ARITH-10)*. June 1991.

178. Williams, T.E. "Analyzing the Latency and Throughput Performance of Self-Timed Pipelines and Rings" in *VLSI-91 IFIP Conference*. August 1991.

179. Williams, T.E. and Horowitz, M.A., "A Zero-Overhead Self-Timed 160-ns 54-b CMOS Divider." Journal of Solid-State Circuits. Vol. **26**(11): pgs. 1651-1661. November 1991. Also appears in the Proceedings of the 1991 International Solid-State Circuits Conference, San Francisco, CA. February, 1991.

180. Williams, T.E. "Self-Timed Rings and Their Application to Division." Ph.D. Thesis from Stanford University, May 1991. Appears as CSL Technical Report No. 91-482.

181. Wingard, D.E., Stark, D.C., and Horowitz, M.A., "Circuit Techniques for Large CSEA SRAM's." Journal of Solid-State Circuits. August 1991.

182. Wolf, M. and Lam, M. "An Algorithmic Approach to Compound Loop Transformations" in *3rd Workshop on Languages and Compilers for Parallel Computing*. August 1990.

183. Wolf, M.E. and Lam, M.S., "A Loop Transformation Theory and Algorithm to Maximize Parallelism." IEEE Transactions on Parallel and Distributed Systems. October 1991.

184. Wolf, M.E. and Lam, M.S. "Data Locality Optimizing Algorithm" in *Conference on Programming Language Design and Implementation (PLDI)*. ACM Sigplan. Toronto, Canada. June 1991.

185. Wong, D., De Micheli, G., and Flynn, M. "Designing High-Performance Digital Circuits Using Wave Pipelining" in *VLSI Conference*. IEEE. Munich, W. Germany. August 1989.

186. Yang, T., Horowitz, M., and Wooley, B. "A 4nsec, 4Kxlbit Two-Port BiCMOS SRAM" in *Custom Integrated Circuits Conference*. IEEE. Rochester, NY. May 1988.

187. Yang, T.S., Horowitz, M., and Wooley, B., "A 4ns 4Kx1bit Two-Port BiCMOS SRAM." IEEE Journal of Solid-State Circuits. Vol. **SC-23**(5): pgs. 1030-1040. October 1988.